

Power System Extreme Event Screening using Graph Partitioning

Bernard C. Lesieutre, Sandip Roy, Vaibhav Donde, and Ali Pinar

Abstract— We propose a partitioning problem in a power system context that weighs the two objectives of minimizing cuts between partitions and maximizing the power imbalance between partitions. We then pose the problem in a purely graph theoretic sense. We offer an approximate solution through relaxation of the integer problem and suggest refinement using stochastic methods. Results are presented for the IEEE 30-bus and 118-bus electric power systems.

Index Terms—Graph Partitioning, Extreme Events

I. INTRODUCTION

THIS work is motivated by extreme events in electric power systems that are caused by multiple contingencies. Anticipation of such events is made difficult by the fundamentally combinatorial nature of the problem. In this paper we study a class of extreme events that may occur in a power network – the partitioning of a network into separate areas with a disturbance severity measured as the power flow disrupted by the separation – using a graph-theoretic approach.

In practice the power grid is operated to be at least “N-1” secure, where N-1 refers to the number of components in the model, i.e., a nominal number, N, less a single component. For example, in the WECC Operating Handbook this famous N-1 criterion is stated as “the interconnected power system shall be operated at all times so that general system instability, uncontrolled separation, cascading outages, or voltage collapse, will not occur as a result of any single contingency or multiple contingencies of sufficiently high likelihood” [1]. More specific operating policies are then employed to achieve this goal.

The N-1 criterion is seemingly practical. In planning stages, engineers may use traditional tools to prepare for virtually all single contingencies and a number of more likely double and triple contingencies. Furthermore, we may expect that multiple random outages are unlikely, and we might argue they are of little concern. However, it is observed that the probabilities of component outages are not independent. A single contingency

may initiate a subsequent outage or at least serve as an indicator that additional outages may occur. A dramatic example of dependent outages occurred during the August 14th, 2003 blackout in which the initiating events were multiple lines sagging into the trees. The common cause of failure of these lines suggests that their failures were not independent events. Furthermore, the word “security” has taken on an additional meaning in recent years, and we are concerned about malicious attacks on the power grid in which outages are entirely dependent.

The mathematics involved in a brute-force enumeration of multiple contingencies is not practical, not even using supercomputers. This is easily demonstrated by direct calculation of the number of combinations required for N-3 and N-4 analysis for a 10,000 component system. The N-3 analysis involves approximately $10^{12}/6$ combinations and the N-4 analysis involves $10^{16}/24$. Assuming a supercomputer with 10,000 parallel processors and assuming that each processor can evaluate one combination in 1 second, the computation times will exceed 193 days and 10^3 years respectively. For a 100,000 component system the computing times would exceed 528 years and 10^7 years respectively. Clearly detailed analysis of all possible multiple contingencies is not computationally feasible.

A sensible alternative to brute-force enumeration is to search intelligently for the contingencies that may lead to extreme events. In this paper we examine a reduced objective and simplified techniques for analysis that aid in the search for extreme events. First, we limit the scope of our search to focus on events with severe consequences that may arise from relatively few component outages. The explicit trade-off between these two factors is made explicit in our objective model. Second, we employ a graph theoretic approach to find system partitions arising from a small number of component outages that result in a large disruption in power flow between the partitions. This is a measure of severity.

This approach should be used as a screening step after which a more detailed analysis using appropriate power system models can be employed to evaluate outages of subsets of the identified components that may result in a severe event. In this way we can hope to identify contingencies that can cause severe outages without exhaustive search. We demonstrate in our prior work that this general approach does identify outages [2], [3]; we do not discuss the detailed-analysis step further here. We limit this paper to the problem of finding undesirable partitions (causing severe power imbalance) with the least effort (fewest cuts).

B.C. Lesieutre, V. Donde and A. Pinar are supported on this work by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of U.S. Department of Energy under contract DE-AC03-76SF00098.

B.C. Lesieutre, V. Donde and A. Pinar are with the Lawrence Berkeley National Laboratory, Berkeley, CA 94720 USA (e-mails: BCLesieutre@lbl.gov, VDonde@lbl.gov and APinar@lbl.gov).

Sandip Roy is with Washington State University, Pullman, WA (e-mail: sory@eeecs.wsu.edu).

In the next section, we provide basic background on spectral partitioning of graphs, since our analysis closely follows methods employed in this field. In Section III we formally state our problem and pose a solution technique, and in Section IV we provide illustrative examples. We conclude with a discussion of the benefits and weaknesses of this approach.

II. GRAPH PARTITIONING – SPECTRAL METHOD

This section is intended to be a tutorial and may be skimmed by those readers already familiar with this topic. In keeping with the spirit of a tutorial we introduce terminology and concepts in reference to a small sample graph depicted in Figure 1.

The four nodes and five branches are represented by circles and arrows respectively. This graph is “directed” in the sense that the arrows start at one node and point to another. A branch/node incidence matrix, A , that mathematically represents the information in the graph (nodes and directed connections) is given by

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad (1)$$

where the rows represent the branches and the columns represent the nodes. Matrix entry A_{mn} is equal to 1 if branch m originates at node n , and it is equal to -1 if the branch terminates (points to) node n . The entry is equal to zero if the branch is not incident (connected) to node n .

A. Graph Partitioning

The purpose of graph partitioning is to separate the nodes of a graph into two or more groups to satisfy a specified goal. In the power system context, this may be familiar to the reader in the “slow coherency” literature. Groups of generators are said to be coherent when they have identical dynamic responses to

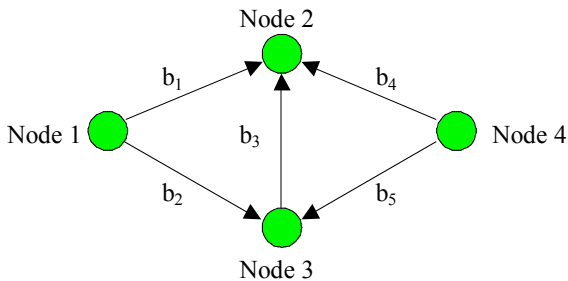


Fig. 1. A small 4 node, 5 branch reference graph.

events originating outside their group. Theory supporting this, as well as certain techniques for identifying such groups, rely on identifying weak cutsets separating groups in the network.

In computer and computational contexts, it is often the goal to form equal sized groups with the minimum number of edges crossing between different parts. It is commonly used as part of the divide-and-conquer paradigm to decompose problems into loosely coupled sub-problems. Three main areas that have driven the research on graph partitioning are parallel processing, sparsity preserving orderings for sparse matrix factorizations, and VLSI design. In parallel processing, the vertices represent computational tasks, and edges correspond to communication requirements between two tasks. Vertices are weighted with the estimated computational times of the associated tasks, and thus balance among total weights of parts grants computational load balance among processors. Minimizing the edge cut on the other hand, minimizes the total communication overhead in the system. In sparse matrix ordering, graph partitioning is used as the decomposition operator for nested dissection approach, which is an example of the divide-and-conquer paradigm. The performances of these approaches are determined by the sizes of the separators, and thus it is crucial to find good separators, that can decompose the problem into at least two smaller subproblems. Finally in VLSI design, partitioning approaches are used to decompose the problem into smaller tractable problems, again as part of the divide-and-conquer paradigm. For example, vertices can be used to represent circuit components, and edges represent circuit connections. A good layout can be achieved by decomposing this graph into smaller subgraphs, which give tractable size problems.

As we have seen in these practical examples, common criteria include minimizing the number of lines separating the groups (min cut), minimizing the cuts while creating groups of equal size (bisection), and more. Mathematically it is convenient to introduce an indicator vector associated with the nodes, the elements of which are assigned a value that is unique to each group. For two groups the values are typically chosen to be -1 and 1 . For example, if we choose to place nodes 1 and 2 in one group and nodes 3 and 4 in another, we can specify the partition with the indicator vector

$$x = [-1 \quad -1 \quad 1 \quad 1]^T. \quad (2)$$

Multiplying the incidence matrix by the indicator vector yields a vector that identifies the branches that separate the two groups:

$$y = Ax = \begin{bmatrix} 0 \\ -2 \\ 2 \\ 2 \\ 0 \end{bmatrix}. \quad (3)$$

The value of an entry in vector y is 0 if the branch is internal to a group and is ± 2 if the branch connects the two groups. The sign depends on the direction of the arrow. From (3) it is obvious that

$$y^T y = x^T (A^T A) x = 4 \text{ times \# of separating branches.} \quad (4)$$

We will use this simple expression relating the incidence matrix, the indicator vector, and the number of cuts separating

two groups shortly. First we comment on the matrix $A^T A$ that appears in (4). This matrix is called the “Laplacian” matrix associated with a graph and has the following properties:

1. It is square and symmetric with dimension equal to the number of nodes.
2. The off-diagonal entries are either 0 or -1 .
3. The diagonal entry in each row is equal to the negative of the sum of the off-diagonal entries.
4. It is independent of the orientation of the branches in the graph.
5. If the graph is connected, it has a single zero eigenvalue with a corresponding eigenvector equal to a vector with identical entries – typically expressed as vector of all ones.¹ (Note the rows of A sum to zero.)
6. Its other eigenvalues are all positive. (The matrix is obviously positive semi-definite from (4) in which $y^T y \geq 0$ for all x .)
7. The signs of the entries of the eigenvector associated with the smallest positive eigenvalue serve as an indicator vector that partitions the graph into two internally connected groups.

Properties 1-6 in this list are easily verified. Property 7, which is less obvious, is attributed to Fiedler [4], and the smallest positive eigenvalue of the Laplacian matrix and corresponding eigenvector are often referred to as the Fiedler eigenvalue and Fiedler eigenvector. These play a critical role in the spectral graph partitioning method we review in the next subsection. For reference, the Laplacian matrix for our example system is given by

$$L = A^T A = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}. \quad (5)$$

The reader may verify the properties listed above for this matrix.

B. Graph Partitioning – Spectral Approach

As discussed earlier, in many applications of graph partitioning, the goal is to partition a graph into two even parts so as to minimize the number branches connecting the parts. Denoting the Laplacian matrix by $L = A^T A$ and using (4) we can cast this problem as an optimization of a quadratic cost over all possible partition indicator vectors:

$$x^* = \arg \min_{x_i \in \{-1, 1\}} x^T L x. \quad (6)$$

This is an integer programming problem. Though not NP-complete (see Stoer [5] for a clever deterministic algorithm), it can be computationally costly to solve. One approach to quickly find a result is to “relax” the values of x to allow them to take on real values, i.e., to find

¹ If the graph comprises disconnected subgraphs, then they may be treated separately and the number of zero eigenvalues equals the number of disconnected subgraphs.

$$\hat{x} = \arg \min_{x_i \in \mathbb{R}} x^T L x \quad (7)$$

and then obtain the partition according to the sign pattern of x . We note that this formulation admits the trivial solution $x=\mathbf{1}$ (vector of all ones). The trivial solution does not partition the graph since the indicator vector places all nodes in the same group! Excluding this solution temporarily (i.e., considering only the solutions orthogonal to the trivial solution), the next best vector is the Fiedler eigenvector since it

- minimizes (7) subject to a scaling, or more precisely, it minimizes

$$\frac{x^T L x}{x^T x} \text{ where } x \perp \mathbf{1}$$

- and finds a partition whose two subgraphs are connected.

Note that this spectral solution of (7) is not guaranteed to find the optimal solution for (6), but it works well in practice. We can extend this technique to the bisection problem, the problem of breaking the graph into two equally sized partitions. To do so, we add the trivial solution to the Fiedler eigenvector. Denoting the Fiedler eigenvector by v_F , consider

$$x = v_F + \alpha \mathbf{1}. \quad (8)$$

By choosing α to equalize the number of positive and negative entries in x , and partitioning the graph according to the sign pattern of x , we obtain a good bisection of the graph.

Spectral partitioning methods have been successfully applied to several partitioning problems. Pothen, Simon and Liou showed the first results for applying spectral methods to graph partitioning [10]. Spielman and Teng showed that provably good solutions can be achieved for partitioning finite elements meshes using spectral methods [18]. Pothen et al. applied spectral methods to nested dissection ordering for sparse matrices [11]. In the VLSI domain, Alpert and Yao showed how multiple eigenvectors can be used for better results [17]. More recently, spectral methods gave way to multilevel graph partitioning methods initiated by Hendrickson and Leland [14]. Today tools based on multilevel graph partitioning such as Chaco [14], Metis [15], Party [16], Jostle [13], and Scotch [12] are widely used. However, spectral graph theory remains as a productive field for its flexibility to model a wide variance of problems, as we will see in this paper. The reader is referred to [6] for a survey on graph partitioning methods, including spectral methods.

III. PROBLEM FORMULATION

The underlying power network may be described by a graph. Network buses (nodes) are connected by transmission lines (branches); power is injected at certain nodes and extracted at others. The focus of this paper is to identify a small number of lines that if removed will partition the system **and** will cause a severe power imbalance. These are two, possibly competing, objectives:

1. Minimize cuts

2. Maximize directed power flow across the cutset.

Surprisingly there have been few efforts to address this flow-based partitioning problem. Standard graph partitioning tools are constrained by forcing balance, and cannot support the producer/consumer roles for vertices as in the case of our problem. The max-flow/min-cut problem is similar and well-known as one of the fundamental problems of combinatorial algorithms, but it is different than the problem posed here – despite the similarity in and usage of descriptive words! Here, the imbalance between the two parts is equally important as the cutsize, and its addition moves some versions of our problem to the NP-complete status, whereas maximum-flow algorithms can be solved efficiently in polynomial time. The complexity of our problem is sensitive to how the objective function is stated. We are currently working on our analysis of the complexities of various objective functions, and we postulate that the problem is NP-complete for two of the three versions considered here, and for many versions of the problem.

To proceed we first describe these objectives in terms of an indicator vector that will describe the partitions. The cutsize cost was expressed in the previous section by (4) and (6). To obtain an expression for the power flows across a specific cutset of lines we might employ a power flow program to calculate the line flows given the power injections. By assuming a lossless network, as an approximation, we can avoid this added complexity. Let the vector p denote the bus power injections and let x be an indicator vector with entries equal to ± 1 . Then,

$$p^T x = 2 \text{ (power flow between groups)}. \quad (9)$$

To see why this is true, we note that the sum of power injections in the generation-rich partition must be equal to the power flow from that group to the generation-poor partition, for a lossless system.

There are a number of ways (9) and (6) can be applied to approach our problem. We consider three in this paper.

A. Simple cuts less power imbalance

The simplest, and arguably the most direct, mathematical representation for the objectives stated above is given by the following minimization problem:

$$x^* = \arg \min_{x_i \in \{-1,1\}} x^T L x - c p^T x \quad (10)$$

where the first term is equal to 4 times the number of cuts, the second term is proportional to the power flow disrupted by such cuts, and the positive scalar c explicitly represents the trade-off between these objectives. A small value for c favors minimizing cuts, while a large value of c favors maximizing power imbalance.

To find a solution to the partitioning problem (10), we can use an approach similar to that used for spectral partitioning in

the previous section². In particular, we relax x in (10) to allow for any real valued entries:

$$\hat{x} = \arg \min_{x_i \in \mathbb{R}} x^T L x - c p^T x. \quad (11)$$

Direct application of calculus yields

$$2Lx - cp = 0. \quad (12)$$

with the following solution achieving the minimum of (11)

$$x = \frac{c}{2} L^\dagger p + \alpha 1 \quad (13)$$

where the symbol “ \dagger ” is used to represent the pseudo inverse. To understand the solution in (13), it is critical to note that vector p is orthogonal to the vector of all ones (since the elements of p sum to zero), and thus lies in the column space of L . The family of solutions parameterized by α in (13) satisfy (12) exactly.

To find a useful discrete indicator vector from the continuous solution (13), we recommend examining the $n-1$ forms taken by the vector

$$x = \text{sign}\left(\frac{c}{2} L^\dagger p + \alpha 1\right), \quad (14)$$

as the parameter α is varied from a large negative number to a large positive number, ignoring the cases when all elements are either positive or negative. (These indicate all nodes in a single group.) From *these* $n-1$ possibilities, we recommend the choice of the indicator vector that minimizes the objective (10). As we will discuss through examples in Section V, this approach is not perfect, but it is simple and yields useful results.

Substitution of (13) into (11) provides a lower bound on the minimization as a function of c , p , and L :

$$x^T L x - c p^T x \geq \frac{-c^2}{4} p^T L^\dagger p \quad (15)$$

This bound is potentially important for two reasons. First, when we restrict elements of x to take on values equal to ± 1 to create an indicator vector, (15) allows a comparison to the bound. This can be used to gauge the quality of the integer indicator vector solution. Secondly, the structure of the bound provides insight into robust design for a power grid in terms of the network structure and the power injections. The most robust system, requiring many outages to create a substantial disturbance, will have the eigenvector associated with the largest eigenvalue of L , aligned with the power injection vector to minimize the quantity $p^T L^\dagger p$.

Continuing with our tutorial example introduced in the previous section, let us define the power injection vector

$$p = [2 \quad -2 \quad 1 \quad -1]^T. \quad (16)$$

The $n-1=3$ candidate indicator vectors suggested by (15) are

² This formulation of the problem can be solved in polynomial time, and relatively quickly in practice. Here, for consistency with the remaining two formulations, we report on the use of the spectral approach.

$$x_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}. \quad (17)$$

For values of c less than 2, the minimum objective is obtained with x_3 . For values of c greater than 2, the minimum objective is obtained with x_2 . These are clearly the correct solutions of (10) for this example. For the small c case, we place more emphasis on small cuts. The x_3 solution contains the fewest cuts possible, 2, and has a greater power imbalance than x_1 , which also has 2 cuts. For large c , we emphasize the power imbalance more, and the indicator vector given by x_2 maximizes this quantity in this example.

B. Simple cuts less the square of power imbalance

The objective (10) presented in the previous subsection is only one of the many possible objectives that trade off line outages and power imbalance. We will consider another in this section, but pause to mention two peculiarities with the results in the prior section. First, the candidate indicator vectors constructed from continuous solution (13) do not depend on the trade-off parameter c . The choice of indicator vector, following the suggested procedure, does depend on this parameter, but the set of $n-1$ candidate vectors does not. Second, the bound provided in (15) is not useful for large values of c . The issue is that the indicator vectors have a constant size, i.e., $x^T x = n$. The unconstrained vectors in (13) that are used to calculate the bound in (15) may be arbitrarily large, especially for large c , and do not serve to evaluate the quality of a solution in these cases. One can find an approximate lower bound (which works reasonably well) by similarly rescaling the solution vector x as expressed in (13), but the result is only approximate, and might not be a true bound. (To properly follow this line of reasoning, one would constrain the optimization problem to require $x^T x = n$, the same magnitude as any indicator vector, and the result can be used to calculate a true bound.³)

In the absence of any compelling reason to restrict our studies to the objective posed in (10) we examine the slightly different problem

$$\begin{aligned} \arg \min_{x_i \in \{-1, 1\}} x^T L x - c x^T p p^T x \\ = \arg \min_{x_i \in \{-1, 1\}} x^T (L - c p p^T) x \end{aligned} \quad (18)$$

In (18) the power imbalance term is squared. The form of (18) is interesting and mathematically convenient. The matrix $(L - c p p^T)$ has a similar form as a (weighted) Laplacian matrix. It exhibits a trivial structural zero eigenvalue with corresponding eigenvector equal to the vector of all ones.

The solution, subject to a scaling, is easily obtained using the spectral arguments we reviewed in Section II:

$$x = v_l + \alpha \mathbf{1} \quad (19)$$

where v_l is the eigenvector associated with the smallest eigenvalue, excluding the trivial structural eigenvalue. The suggested procedure for choosing a discrete indicator vector is the same as the presented in the previous subsection. In this case, however, a practical closed-form lower bound is obvious:

$$\frac{x^T (L - c p p^T) x}{x^T x} \geq \frac{v_l^T (L - c p p^T) v_l}{v_l^T v_l}. \quad (20)$$

C. Ratio of squared power imbalance to cuts

In formulation (18), a special case occurs when

$$c = \frac{1}{p^T L^\dagger p}. \quad (21)$$

The matrix $(L - c p p^T)$ exhibits a nontrivial zero eigenvalue with corresponding eigenvector

$$v_l = L^\dagger p \quad (22)$$

This special solution is particularly interesting because it also maximizes the ratio of squared power imbalance to cut lines cost, and it may itself be considered a reasonable objective function:

$$\max_{x_i \in \{-1, 1\}} \frac{x^T p p^T x}{x^T L x}. \quad (23)$$

While the relaxations of (23) and (18) yield identical results, the integer forms may not. Also, as we will see in one of our examples, when indicator vectors are chosen following the simple procedure described in this paper, different results may be obtained.

Direct substitution of (22) into (23) yields a practical upper bound for this ratio:

$$\frac{x^T p p^T x}{x^T L x} \leq p^T L^\dagger p. \quad (24)$$

Again we note the interpretation for this bound suggests that less likely extreme events or less severe extreme events may be achieved by designing the system such that the eigenvector associated with the largest eigenvalue of the Laplacian is aligned with the injection vector p .

Examination of the tutorial example system yields the same partitions discussed in the previous subsections.

IV. EXAMPLES

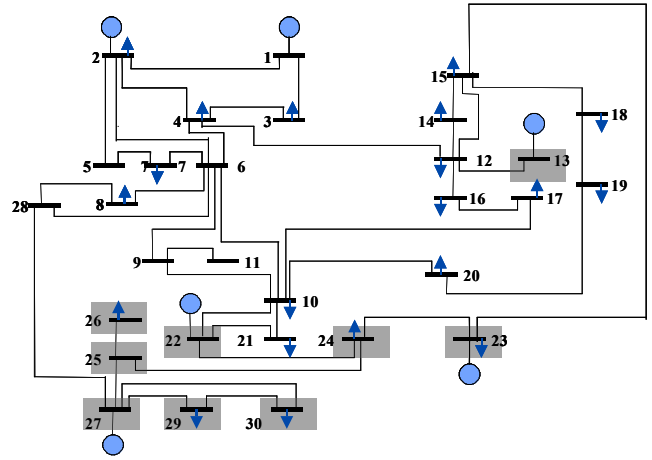
In this section we examine two test systems: the IEEE 30 bus and the IEEE 118 bus systems [7]. These are convenient because they are readily available and many researchers are familiar with them. We present the 30-bus system results in detail, and discuss the 118 bus results.

³ Numerically, one can easily compute such a solution to provide a more accurate lower bound.

The 30 bus system is shown in Fig. 2. The set of power injections used in this system were modified from the original data set to increase power flows and hence the imbalance in the network. (The original data contains 3 power-balanced areas.) The power injection data used is provided in the appendix. The value for the trade-off parameter c is specified as in (21). This allows us to compare results using the three optimization goals discussed in the Section III, (10), (18), and (23).

TABLE 1. COMPARISON OF OBJECTIVE BOUNDS TO CALCULATIONS USING THE CALCULATED INDICATOR VECTOR AND A RANDOM INDICATOR VECTOR FOR THE 30-BUS SYSTEM.

Next we examine the IEEE 118 bus system. In this case the challenge of going from a relaxed continuous solution to a discrete indicator vector becomes apparent. Objective functions A and B yield a different partition from Objective C, using the approach suggested in previous sections and choosing c as defined in (21).



that is near to the relaxed solution. The continuous problem also provides bounds for the integer solution.

These bounds cannot be achieved, except in the most contrived of cases. The elements of the indicator vector are restricted to take on the values -1 or 1 . If there are n nodes, then the possible forms the vector may take may be geometrically described as the vertices of an n -dimensional hypercube. In contrast, the continuous approximation may be considered as a vector pointing anywhere in an n -dimensional space. Only when this vector is pointing directly at a vertex will this bound be met. Generically, this will not occur.

One can use similar arguments to explain why the nearest vertex to the continuous solution vector may not yield the optimal results. Moving away from the optimal solution in any direction (except that aligned with the vector of all ones, which is accounted for in this paper's calculation), will necessarily cause the objective function to evaluate further from the bound, but some directions are better/worse than others. It is entirely possible that the nearest vertex is in a costly direction while a more distant vertex corresponds to little change in the objective function.

We are continuing our work in several directions. We are working on evaluating the objectives presented here and we seek other sensible objectives. We are developing refinement algorithms to aid in locating better choices for indicator vectors from the relaxed solutions for Objectives B and C. (Objective A can be solved exactly using other techniques.) We are seeking to employ directional information as mentioned above, and we are also experimenting with stochastic refinement algorithms motivated by randomized min-cut algorithms [8], by the influence model [9], and a randomized modification to Stoer's algorithm.

Finally, we are examining explicit use of the bound (24) as a metric for comparing the relative "brittleness" of different grids, and different operating conditions. We are also examining the use of this bound for quickly assessing the relative merits of line additions at specific locations. We will report on these findings in a separate publication.

APPENDIX

Table 2 lists the power injections that we used for the 30 bus example shown in Fig. 2. The values are expressed in MW; calculations in the paper use per-unit values with the power base equal to 1000MW.

ACKNOWLEDGMENT

The authors thank Virgil Rose of the California Institute for Energy and the Environment for pointing out that the bounds we derived could be interpreted as a measure of system brittleness, or susceptibility to extreme events.

2	-7.3	12	-112	22	315.9
3	-24	13	410	23	310
4	-76	14	-62	24	-87
5	0	15	-82	25	0
6	0	16	-35	26	-35
7	-228	17	-90	27	469.1
8	-300	18	-32	28	0
9	0	19	-95	29	-24
10	-58	20	-22	30	-106

REFERENCES

- [1] Western Electricity Coordinating Council, Operating Committee Handbook, p. III-119, revised 2005.
- [2] V. Donde, V. López, B.C. Lesieutre, A. Pinar, C. Yang, and J. Meza, "Identification of severe multiple contingencies in electric power networks," in *Proceedings of the North American Power Symposium*, Ames, IA, October 2005.
- [3] V. Donde, V. López, B.C. Lesieutre, A. Pinar, C. Yang, and J. Meza, "Identification of Severe Multiple Contingencies in Electric Power Systems," submitted to the *IEEE Transactions on Power Systems*.
- [4] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory," *Czechoslovak Mathematical Journal*, 25(100), pp.619-633, 1975.
- [5] M. Stoer and F. Wagner, "A Simple Min-Cut Algorithm," *Journal of the ACM*, Vol. 44, No. 4, July 1997, pp. 585-591.
- [6] B.L. Chamberlain, "Graph partitioning algorithms for distributed workloads of parallel computations," Tech Report TR-98-10-03, University of Washington, Dept. of Computer Science and Engineering, October 1998.
<http://www.cs.washington.edu/homes/brad/cv/pubs/degree/generals.pdf>
- [7] Power System Test Case Archive,
<http://www.ee.washington.edu/research/pstca>
- [8] D. Karger, "Randomization in Graph Optimization Problems: A Survey," *Optima*, No 58, June, 1998.
- [9] C. Asavathiratham, S. Roy, B.C. Lesieutre, and G.C. Verghese, "The Influence Model," *IEEE Control Systems Magazine*, pp.52-64, December 2001.
- [10] A. Pothen, H. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM Journal on Matrix Analysis and Applications*, 11(3): 430-452, 1990.
- [11] A. Pothen, E. Rothberg, H. Simon, and L. Wang, "Parallel sparse Cholesky factorization with spectral nested bisection ordering," *Technical Report RNR-94-011, Numerical Aerospace Simulation Facility, NASA Ames Research Center*, May 1994.
- [12] F. Pellegrini, "SCOTCH 4.0 User's guide," *Research report Laboratoire Bordelais de Recherche en Informatique*, 2006.
- [13] C. Walshaw, M. Cross, and K. McManus, "Multiphase Mesh Partitioning. Appl. Math. Modelling," 25(2):123-140, 2000.
- [14] Bruce Hendrickson and Robert Leland, "The Chaco User's Guide: Version 2.0," Sandia Tech Report SAND94-2692, 1994.
- [15] George Karypis and Vipin Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *International Conference on Parallel Processing*, pp. 113-122, 1995.
- [16] R. Preis and R. Diekmann, "The PARTY Partitioning-Library, User Guide - Version 1.1," *Technical Report tr-rsfb-96-024, University of Paderborn*, Sep. 1996.
- [17] C. J. Alpert and S. Z. Yao, "Spectral Partitioning, the More Eigenvectors the Better," *32nd ACM/IEEE Design Automation Conference*, San Francisco, June 1995, pp. 195-200.
- [18] D.A. Spielman and S.H. Teng, "Spectral Partitioning Works: Planar graphs and finite element meshes," *Proc. IEEE Symposium on Foundations of Computer Science*, pp. 96-105, 1996.

TABLE 2. POWER INJECTIONS USED FOR THE 30-BUS EXAMPLE.

Bus	P	Bus	P	Bus	P
1	145.3	11	0	21	-175